

## ADAPTIVE HYBRID PRISMATIC-TETRAHEDRAL GRIDS FOR VISCOUS FLOWS

Yannis Kallinderis<sup>\*</sup>, Aly Khawaja<sup>†</sup>, and Harlan McMorris<sup>‡</sup>

Dept. of Aerospace Engineering and Engineering Mechanics

The University of Texas at Austin

Austin, TX 78712

### ABSTRACT

The paper presents generation of adaptive hybrid prismatic/tetrahedral grids for complex 3-D geometries including multi-body domains. The prisms cover the region close to each body's surface, while tetrahedra are created elsewhere.

Two developments are presented for hybrid grid generation around complex 3-D geometries. The first is a new octree/advancing front type of method for generation of the tetrahedra of the hybrid mesh. The main feature of the present advancing front tetrahedra generator that is different from previous such methods is that it does not require the creation of a background mesh by the user for the determination of the grid-spacing and stretching parameters. These are determined via an automatically generated octree.

The second development is an Automatic Receding Method (ARM) for treating the narrow gaps in between different bodies in a multiply-connected domain. This method is applied to a two-element wing case.

A hybrid grid adaptation scheme that employs both *h*-refinement and redistribution strategies is developed to provide optimum meshes for viscous flow computations. Grid refinement is a dual adaptation scheme that couples division of tetrahedra, as well as 2-D directional division of prisms.

### INTRODUCTION

Simulation of flows around three-dimensional bodies is a major issue in computational fluid mechanics. Geometrical and flow-field complexity combine to make 3-D computations a pacing item. The generation of a body-conforming grid has proven to be a difficult task [1, 2].

The success of a structured grid generation may be extremely dependent on geometry and operator proficiency. Block-structured schemes exist which, based on extensive user input, break the computational domain into a number of blocks within which hexahedra are constructed. A radical alternative to structured meshes is to use tetrahedra. Tetrahedral grids provide flexibility in 3-D grid generation since they can cover complicated topologies easier compared to the hexahedral meshes. This does not come without a price, viz., unstructured grids require a great deal more memory than their structured counterparts. They

---

<sup>\*</sup>Associate Professor

<sup>†</sup>Graduate Research Assistant

<sup>‡</sup>Graduate Research Assistant

employ pointers to provide connectivity information between cells, faces, edges, and nodes. Additionally, approximately five to six times more tetrahedra than hexahedra are required to fill a given region with a fixed number of nodes. This vast increase in the number of required cells leads directly to impractical memory requirements for 3-D viscous flow simulations. Furthermore, resolution of the strong directional flow gradients encountered in viscous flows requires very thin grid elements. It is very expensive to generate tetrahedral cells with high aspect ratio to resolve such gradients.

One solution to the dilemma between hexahedra and tetrahedra is to use a semi-structured grid made of prisms. Prismatic cells are composed of triangular faces in the lateral (body-surface) directions and quadrilateral faces in the normal direction. Therefore, they can provide the geometric flexibility of unstructured grids as well as the orthogonality and high aspect ratio qualities of structured grids. Results have been obtained using prismatic grids that reveal their suitability for resolving viscous flow phenomena [3, 4, 5]. The prismatic grid requires a set of pointers to define their base triangular mesh combined with a single index for each prism belonging to the same stack [4]. Finally, structure of the prisms in one of the directions can be exploited in order to apply directional multigrid acceleration within the viscous regions [4, 6].

The areas between different prismatic layers covering the surfaces of the domain can be quite irregular. Furthermore, the relevant flow features do not usually exhibit the strong directionality that the viscous stresses have. Tetrahedral elements appear to be appropriate for these irregularly shaped regions. Their triangular faces can match the corresponding triangular faces of the prisms.

The present work employs two families of grid elements: prismatic grid cells for the viscous region and tetrahedral grid cells elsewhere. A new advancing front type of method is developed for generation of the tetrahedra of the hybrid (prismatic/tetrahedral) mesh. The main feature that is different from previous advancing front generators [7, 8] is that it does not require a user-constructed background mesh for determination of the grid-spacing and stretching parameters. It should be noted that generation of the background mesh has been a very time-consuming and user-dependent part of previous advancing front methods. A special octree is constructed via a *Divide-and-Conquer* method of the space outside of the region covered by the prisms. The grid spacing is then determined based on the size of local octants which form the octree.

An important issue arising with use of semi-structured prismatic grids is covering of narrow gaps in between different bodies. In the present work, an Automatic Receding Method (ARM) is developed that relies on receding the prisms layers that surround each body. The gap that the prisms leave is filled with tetrahedra. The case of a two-element wing is considered as a test of the technique.

A High Speed Civil Transport (HSCT) type of aircraft geometry is considered in order to investigate efficiency and to demonstrate robustness of the method in handling relatively complex topologies. The generated hybrid grid required only 170 K tetrahedra instead of an estimated two million had a tetrahedral mesh been used in the prisms region, as well.

A hybrid grid adaptation scheme that employs both  $h$ -refinement and redistribution strategies is developed to provide optimum meshes for viscous flow computations. Grid refinement is a dual adaptation scheme that couples division of tetrahedra, as well as 2-D directional division of prisms.

## GENERATION OF PRISMS

An unstructured triangular grid is employed as the starting surface to generate a prismatic mesh. This grid, covering the body surface, is marched away from the body in distinct steps, resulting in generation of semi-structured prismatic layers in the marching direction. The goal of the marching scheme is to reduce

the curvature of the previous marching surface at each step while ensuring smooth grid spacing to avoid surface overlap. The process can be visualized as a gradual inflation of the body's volume. There are three main aspects of the algebraic grid generation process: (i) determination of the directions along which the nodes will march (marching vectors), (ii) determination of the distance by which the nodes will march along the marching vectors, and (iii) smoothing operations on positioning of the nodes on the new layer.

### Determination of the Marching Vectors

Each node on the marching surface is advanced along a marching vector. The marching direction is based on the *node-manifold*, which consists of the group of faces surrounding the node to be marched. The primary criterion to be satisfied when marching is that the new node should be visible from all the faces on the manifold.

The node-normal vector lies on the bisection plane of the two faces on the manifold that form the wedge with the smallest angle. Its location on this plane is determined by bisecting the visibility region on the plane. This process has yielded consistently valid normal vectors at the nodes by constructing the vector most normal to the most acute face planes. Essentially, it does this by maximizing the minimum angle between the node-normal and all the surrounding face normals. A more detailed description of the marching procedure can be found in [3].

Since the visibility requirement is a necessary one in order to obtain a valid grid, all subsequent smoothing operations performed on the original normal vector enforce the visibility constraint.

### Marching Step Size

Determination of marching distances is based on the characteristic angle  $\beta_{ave}$  of the manifold of each node to be marched. This angle is computed using the average dot product between the pairs of faces forming the manifold. The marching distance is a linear function of  $\beta_{ave}$ . It yields relatively large marching distances in the concave regions, and small distances in the convex areas of the marching surface. Specifically, the distance  $\Delta n$  is:

$$\Delta n = (1 + \alpha)\Delta n_{ave}, \quad (1)$$

where  $\Delta n_{ave}$  is the average marching step for that layer, and  $\alpha$  is a linear function of the manifold angle  $\beta_{ave}$ . The sign of  $\alpha$  is positive for concave regions and negative for convex regions.

The average marching step for each layer,  $\Delta n_{ave}$  is computed based on a user specified initial marching step  $\Delta n_o$  and a stretching factor  $st$ . The actual marching step for layer  $j$  is given by:

$$\Delta n_{ave}^j = \Delta n_o \times st^{(j-1)}. \quad (2)$$

### Smoothing Steps

The initial marching vectors are the normal vectors. However, this may not provide a valid grid since overlapping may occur—especially in regions of the grid with closely spaced nodes. To prevent overlapping, the directions of the marching vectors must be altered. A number of smoothing passes (typically 5) are performed over all the nodes on the marching surface. Weighted Laplacian smoothing is applied to the direction of the marching vectors as follows:

$$\vec{V}_i = \omega \vec{V}_i' + \frac{(1 - \omega)}{\sum_j 1/d_{ij}} \sum_j (1/d_{ij}) \vec{V}_j, \quad (3)$$

where  $\vec{V}'_i$  and  $\vec{V}_i$  are the initial and final marching vectors of node  $i$ , while  $\vec{V}_j$  are the marching vectors of the surrounding nodes  $j$  that belong to the manifold of node  $i$ . The weighting factor  $\omega$  is a function of the manifold characteristic angle  $\beta_{ave}$ . It has small values in concave regions, and relatively large ones in convex areas. The averaging of the marching vectors of the neighboring nodes is distance-weighted with  $d_{ij}$  denoting the distance between nodes  $i$  and  $j$ .

A similar procedure is employed for the smoothing of the marching steps  $\Delta n$ .

### Constraints Imposed to Enhance Quality

Typical Navier-Stokes integration methods impose restrictions on the spacing of the points along the marching lines, as well as on smoothness of these lines. In other words, the prismatic grid should not be excessively stretched or skewed. In the present work, two constraints are imposed on the positioning of points on the marching lines, as well as on the deviation of the direction of the marching vectors from one layer to the next. The ratio of the grid spacings along the marching lines of any two consecutive prisms layers should be less than a stretching factor  $st$ :

$$(1 - st)\Delta n|_{j-1} < \Delta n|_j < (1 + st)\Delta n|_{j-1}, \quad (4)$$

where the subscripts  $j - 1$  and  $j$  denote the order of points along each marching line. A typical value of the allowed stretching factor  $st$  is 0.2. Furthermore, the angle between two consecutive marching vectors  $\vec{V}_{j-1}$  and  $\vec{V}_j$  should be less than a specified angle of  $30^\circ$ .

### Scalability of Prisms Generation Time

Grid generation time depends on the number of boundary faces, as well as on the number of prisms layers that are created. The cases of mesh generation around an ONERA M6 wing and an HSC configuration are employed in order to study how the time scales with these two factors. Three different surfaces triangulations of the M6 wing are considered. The first consists of 3239 faces, the second of 8807 faces, while the third has 15279 faces. Figure 1 shows almost linear increase in generation time with number of boundary faces. The number of generated prisms layers was kept the same in all three cases and equal to 24. Perfect linear scaling of the mesh generation time with number of prisms layers is demonstrated in Figure 2. This implies that the required operations of the generator are exactly the same for each layer.

## OCTREE/ADVANCING FRONT TETRAHEDRA GENERATION

A new method is presented for generating the tetrahedra of the hybrid grid. Advancing front type of methods require specification by the user of the distribution of three parameters over the entire domain to be gridded. These field functions are: (i) the node spacing, (ii) the grid stretching, and (iii) the direction of the stretching. In the present work these parameters do not need to be specified. The distribution of grid size, stretching, and direction of stretching is automatically determined via an octree. There is no need for a special background mesh which has been the backbone of previous advancing front generators.

The tetrahedra that are generated should progressively become larger as the front advances away from the original surface. Their size, the rate of increase of their size, as well as the direction of the increase are given from an octree consisting of cubes which is generated automatically via a *Divide-and-Conquer* method. This process generates octants that are progressively larger with distance away from the body. Their size will be the characteristic size of the tetrahedra that will be generated in their vicinity.

Generation starts from the outermost surface of the layer of prisms surrounding the body. The triangles of this surface form the initial front. Then, a list of points is created that consists of a new node, as well as of “nearby” existing points of the front. One of these points is chosen to connect to the vertices of the face. Following choice of the point to connect to, a new tetrahedron is formed. The list of the faces, edges, and points of the front is updated by adding and/or removing elements. The algorithm followed in the present work is the one presented in [8, 9]. The method requires a data structure which allows for efficient addition/removal of faces, edges and points, as well as for fast identification of faces and edges that intersect a certain region. The alternating digital tree (ADT) algorithm is employed for these tasks.

### A Special Octree for Tetrahedral Grid Spacing Control

The *divide-and-conquer* process starts with a *master* hexahedron that contains the body. This hexahedron is recursively subdivided into eight smaller hexahedra called *octants*. Any octant that intersects the body is a *boundary octant* and is subdivided further (inward refinement). The subdivision of those boundary octants ceases when the size of the boundary octant matches the thickness of the prisms on the outermost prisms surface.

Then, the hexahedral grid is further refined in a *balancing process* (outward refinement) to prevent neighboring octants whose depth differs by more than one (*interface octants*). Outward refinement is performed to ensure that the final octree varies smoothly in size within the areas not covered by the prisms. The sole criterion for outward refinement is a depth difference greater than one between the octant itself and any of its neighbors. Only interface octants are subject to outward refinement. Figure 3 illustrates the growth of the size of the octants away from the surface. The figure shows the outermost prisms surface around a High Speed Civil Transport (HSCT) type of aircraft geometry, as well as a cut through the octree. Growth of the octants away from the outermost prisms surface guides growth of the corresponding tetrahedra. Figure 4 illustrates the symmetry plane of the HSCT geometry. The quadrilaterals (dark lines) correspond to the faces of the octants on this plane, while the triangles (light lines) correspond to the faces of the tetrahedra. It is observed that the size of the tetrahedra, as well the stretching of the mesh and the direction of stretching is guided quite accurately by the octree.

Simplicity and no user intervention are main advantages of the octree. The usual trial-and-error procedures for constructing the field functions that give the local size of the tetrahedra, the stretching of the mesh, and the direction of the stretching (background mesh) for previous advancing front generators are avoided in the present method. The octree is generated once and remains the same throughout the generation process.

### Determination of Size of the Tetrahedra

The advancing front method creates a new tetrahedron by connecting each face of the current front to either a new or an existing node. This point is found by using a characteristic distance  $\delta$  which is calculated from the size of the local octant to which the face of the front belongs.

The local characteristic size  $\delta$  is calculated as follows:

$$\delta = Hst^l, \quad (5)$$

where  $H$  is the size of the boundary octants,  $st$  is the stretching parameter, and  $l$  is the level of the local octant to which the face of the front belongs. The values of  $l$  range from 0 (boundary octant) up to a number equal to the number of recursive subdivisions of the initial (master) hexahedron. A typical value

of the stretching parameter  $st$  is 1.8. The smaller the value of  $st$ , the smoother the variation in size of the generated tetrahedra. However, a very small value results in generation of a very large number of elements. The chosen value is a compromise between the two effects. Further details of the method are given in [11].

## THE AUTOMATIC RECEDING METHOD (ARM) FOR HYBRID GRID GENERATION AROUND MULTI-BODY DOMAINS

The developed hybrid grid generation method is flexible and general in order to treat domains that contain multiple bodies. A prismatic layer is created around each one of the bodies, while the regions in between these meshes are filled with tetrahedra. Any location and orientation of these bodies is allowed. This is accomplished via a special method for treatment of narrow gaps that frequently form in multiply-connected domains, such as multi-element wings. The key feature of the method is the fact that the prismatic grid around each of the bodies is generated independently of all the other bodies. As a result, such generation is as simple as the generation of prisms for a domain containing a single body. However, overlapping meshes are avoided here by employing a special technique that redistributes the prisms nodes along their corresponding marching lines after the initial generation. This redistribution occurs in the vicinity of the regions of overlapping prismatic meshes and results in formation of gaps in between the previously overlapping prisms layers. Then a tetrahedral grid is generated in order to fill in those gaps. It should be emphasized that the structure of the prismatic grid is not destroyed.

### Receding of the Prism Layers

Receding of the prisms nodes occurs along the marching lines that intersect with another prismatic mesh. The distance over which the outermost point is “pulled-back” depends on the local extent of overlapping. In order to avoid abrupt changes in the thickness of the layer due to the local receding, the nodes belonging to the neighboring marching lines are also receded to a certain extent. This extent gradually reduces to zero away from the area of overlapping of the meshes. Furthermore, the marching line is not altered due to the rearrangement of the points on it. Finally, redistribution of the points along each marching line obeys the constraint on allowable stretching of the mesh. As a result, the spacing of the first point off the surface is reduced which will have no adverse effect on accuracy of solutions.

The steps that are followed in order to remove overlapping of a specific pair of prism layers are:

1. Find the marching lines of each one of the separate prismatic meshes that intersect with one or more of the other grids.
2. Calculate the length of overlapping of each one of the marching lines of the two or more prismatic meshes that overlap.
3. Redistribute the nodes on these marching lines so that no overlapping occurs. Essentially, this results in receding of the nodes closer to the body surface.
4. Avoid abrupt changes in the thickness of the prisms layers by receding neighboring marching lines that do not intersect. This is accomplished by flagging the neighboring marching lines and calculating the distance of receding ( $\Delta n^r$ ) according to:

$$\Delta n^r = \frac{\sum_{neib}(\Delta n^r)_{neib}}{N} \quad (6)$$

where the subscript *neib* denotes the marching lines that are neighboring the specific line, and *N* is the number of neighboring lines. This lateral smoothing process is repeated a few times (typically 10).

An integral part of the previous method of receding is repositioning of all the points on the marching lines that are 'pulled-back'. A scheme is employed which redistributes the nodes so that the shape of the lines is not altered. In other words, the marching directions are maintained, but the marching distances between consecutive points are modified. This is accomplished by performing a cubic-spline fit to each of the marching lines using the prism node locations for the spline knots. The nodes are then redistributed along the splined lines. The distribution is such that the new node-positions satisfy the grid spacing constraint. In the present work, a certain stretching factor is maintained, while the spacing of the first point off the body surface is reduced.

### Application to Two-Element Wing with Variable Gap Size

In order to illustrate validity of the previous procedure, the case of a two-element wing with variable size of the gap between the main wing and its flap is considered. Figure 5 shows the geometry of the two-element wing. The gap increases along the span. Stage one involves generation of the two separate prismatic meshes that cover each one of the two bodies. Generation is quite simple due to the fact that each layer of prisms is grown independently of the other layer. The two grids overlap locally as shown in Figure 6. In the second stage, the thickness of the prisms layers is reduced locally and the overlap no longer occurs as shown in Figure 7. Comparing the grids of Figures 6 and 7, it is observed that the receding occurs over a larger region which results in a smooth variation of the local thicknesses of both meshes. The final stage involves generation of the tetrahedral mesh that covers all areas in between the prisms. Figure 8 shows the final hybrid (prismatic/tetrahedral) grid on the plane of symmetry. The quadrilaterals are the signature of the prisms on that plane, while the triangles correspond to faces of the tetrahedral mesh.

### HYBRID GRID GENERATION AROUND THE HSCT

A High Speed Civil Transport (HSCT)-type of aircraft geometry was chosen as the test case for the developed grid generator. Figure 9 shows the triangulation of the initial surface. The mesh consists of 4412 triangles and 2275 nodes. A symmetry plane is considered that divides the body. Thus, hybrid grid is generated for half of the space.

The time required to generate the prismatic grid around the HSCT was 90 seconds for 40 layers of prisms on an IBM 390 workstation. Generation of approximately 170,000 tetrahedra took about 67 minutes on the same station. It should be emphasized that employment of a hybrid grid for the HSCT geometry required only 170 K tetrahedra instead of an estimated two million had a tetrahedral mesh been used in the prisms region, as well.

A view of the grown prismatic surface is shown in Figure 10. The growth of the grid is illustrated after 40 marching steps. The effect of the marching process is similar to inflating of the original body volume. Figure 11 shows portion of the initial wing surface, as well as the outer surface of the prisms. Both the structured part (quadrilateral faces), and the unstructured part (triangular faces) of the prisms is shown. Every fourth layer of prisms is shown for clarity of the plot. Two wire-frame views of the prismatic grid are illustrated in Figure 12. The shaded regions correspond to the surface of the aircraft.

Three different stages in the growth of the tetrahedral mesh on top of the prisms is shown in the sequence of Figure 13. The space between the outer prisms surface and the farfield is 'filled up' quite fast.

It is worth observing the transition of the hybrid mesh from the prisms to the tetrahedra. Figure 14 shows the hybrid grid at a section of the symmetry plane of the aircraft. The quadrilateral faces correspond to the prisms, while the triangular faces belong to the tetrahedra. It is observed that the grid transitions smoothly from the prisms region to the tetrahedral area.

The final hybrid grid consists of 176,480 prisms and of 170,300 tetrahedra. Figure 15 shows the signature of the mesh on the symmetry plane. Finally, Figure 16 illustrates the hybrid mesh on two different planes that are perpendicular to each other. The first plane is the symmetry and it is indicated by the darker lines, while the second is intersecting the fuselage at a location upstream of the wing and it is shown via light lines. It should be noted that the irregularity of the lines observed on the second plane are due to the fact that the grid it intersects is not planar as it is on the symmetry plane.

## COMBINED REFINEMENT/REDISTRIBUTION FOR HYBRID GRIDS

A dynamic grid adaptation algorithm has previously been developed for 3-D unstructured grids [10]. The algorithm is capable of simultaneously refining and un-refining appropriate regions of the flow domain. This method is extended to the present work and is coupled with prismatic grid adaptation to implement a hybrid grid adaptation method.

### Directional Division of Prisms

The prisms are refined directionally in order to preserve the structure of the mesh along the normal-to-surface direction. The prismatic grid refinement proceeds by dividing only the *lateral* edges that lie on the wall surface and hence the wall faces. The faces are divided either into two or four subfaces. The resulting surface triangulation is replicated in each successive layer of the prismatic grid. This results in all the prisms that belong to the same stack (namely, the group of cells that originate from the same triangular face on the wall surface) getting divided alike. The prismatic grid refinement preserves the structure of the initial grid in the direction normal to the surface. The primary advantage of using such an adaptive algorithm for prisms is that the data structures needed for its implementation are essentially as simple as that for refining a 2-D triangular grid.

The directional division of the prisms does not increase resolution of flow features that are aligned in a direction that is normal to the wall surface. However, a grid redistribution algorithm can be employed in order to recluster nodes in the normal direction so as to better resolve the viscous stresses [3, 11].

The tetrahedral cells constitute the portion of the grid where inviscid flow features are dominant. These features do not exhibit the directionality that is generally prevalent in viscous stresses. Hence, the tetrahedra are refined by division into eight, four, or two subcells [10].

### Redistribution of Prisms

The redistribution algorithm increases local grid resolution by clustering existing grid points in regions of interest. A measure of the grid resolution required normal to the *no-slip* wall is the values of  $y^+ = \frac{u_\tau y}{\nu}$ , with  $u_\tau = \sqrt{\frac{\tau_{wall}}{\rho_{wall}}}$  being the wall friction velocity. A criterion based on the values of  $y^+$  at the wall is employed to either *attract* nodes towards the wall or *repel* them away from the surface so that a specific value of  $y^+$  is attained at all the wall nodes. This procedure in essence determines a new value for the spacing  $\delta_{wall}$  of the first node off the wall at all locations on the wall surface. The nodes in the prismatic region

are then reclustered along the marching lines emanating from the corresponding wall node, in accordance with the new value of  $\delta_{wall}$ . Details are presented in [4].

### Application of Hybrid-Adaptation Method

The test case of flow past a sphere at a free stream Mach number of  $M_\infty = 1.4$  and a Reynolds number of  $Re = 1000$  (based on the radius of sphere) is considered. The flow is characterized by both inviscid and viscous flow features such as shock waves and boundary layer separation. Details are given in [4].

The hybrid grid adaptation algorithm is implemented to obtain a numerical solution for the flow situation discussed above. A coarse hybrid grid comprising  $\sim 1400$  wall boundary nodes and  $\sim 100K$  tetrahedra is used as the initial grid. The prismatic region is constituted by 20 layers of prisms. The locally adapted grid obtained after  $h$ -refinement based on an initial solution and the Mach number contour lines of the final solution obtained on the adapted grid are shown simultaneously in Figure 17. The figure shows the embedded tessellations on the wall surface, symmetry plane as well as on an equatorial plane cutting through the interior of the grid, normal to the symmetry plane. Mach number contour lines are shown superimposed on the embedded mesh on the equatorial plane in the figure. It is clearly seen that embedding in the tetrahedral region is focussed near the shock location just outside of the prismatic-tetrahedral interface. The prismatic region is also directionally refined near the upstream and downstream sections of the body. This is due to the flow upstream accelerating rapidly from the upstream stagnation point and the flow downstream separating that causes flow gradients in the lateral directions that are detected by the directional adaptive algorithm. The embedded hybrid grid comprises  $\sim 2500$  wall boundary nodes and  $\sim 275K$  tetrahedra. The numerical solution obtained by the solution-adaptive approach is compared with that of the solution obtained on a globally refined grid, starting with the same initial coarse mesh as before. The values of skin-friction coefficients computed on the wall surface are compared between the two solutions, as shown in Figure 18. The figure shows the excellent agreement of the results computed on the locally adapted grid with that of the globally refined grid.

### SUMMARY

Generation of the tetrahedra was made simpler due to the fact that a background mesh was not required for determination of the mesh spacings. The spacing parameters provided by the octree yielded tetrahedral elements that varied in size smoothly. Furthermore, the octree enabled a smooth transition of the grid from the prisms to the tetrahedra.

The time required to generate the prismatic grid around the HSCT type of configuration was 90 seconds for 40 layers of prisms on an IBM 390 workstation. This time scales almost linearly with the number of boundary faces. Generation of approximately 170,000 tetrahedra took about 67 minutes on the same station.

Employment of a hybrid grid for an HSCT-type of geometry required only 170 K tetrahedra instead of an estimated two million had a tetrahedral mesh been used in the prisms region, as well.

The developed method of receding prisms layers (ARM) for narrow gaps between different bodies was validated via a case of a two-element wing with variable size of the gap between the two elements.

## ACKNOWLEDGMENTS

This work was supported by NASA Grant NAG1-1459 and NSF Grant ASC-9357677 (NYI Program). The authors wish to thank Ms. Karen Bibb of NASA Langley for giving us access to the FELISA mesh generation code.

## References

- [1] J. F. Thompson, and N. Weatherill, "Aspects of Numerical Grid Generation: Current Science and Art", AIAA Paper 93-3539-CP, 1993.
- [2] T. J. Baker, "Developments and Trends in Three Dimensional Mesh Generation," *Applied Numerical Mathematics*, Vol. 5, pp. 275-304, 1989.
- [3] Y. Kallinderis and S. Ward, "Prismatic Grid Generation for 3-D Complex Geometries", *Journal of the American Institute of Aeronautics and Astronautics*, Vol. 31, No. 10, pp. 1850-1856, October 1993.
- [4] V. Parthasarathy, Y. Kallinderis, and K. Nakajima, "A Hybrid Adaptation Method and Directional Viscous Multigrid with Prismatic-Tetrahedral Meshes," AIAA Paper 95-0670, Reno, NV, January 1995.
- [5] K. Nakahashi, "Optimum Spacing Control of the Marching Grid Generation", AIAA paper 91-0103 , 1991.
- [6] Parthasarathy, V. and Kallinderis, Y., "Directional Viscous Multigrid Method on Adaptive Prismatic Meshes ," *AIAA Journal*, Vol. 33, No. 1, January 1995.
- [7] R. Lohner, and P. Parikh, "Generation of Three-Dimensional Unstructured Grids by the Advancing-Front Method ," AIAA Paper 88-0515, 1988.
- [8] J. Peraire, J. Peiro, L. Formaggia, K. Morgan and O.C. Zienkiewicz, "Finite Element Euler Computations in Three Dimensions," *AIAA 26th Aerospace Sciences Meeting*, Reno, AIAA Paper 88-0032, January 1988
- [9] J. Peiro, J. Peraire, and K. Morgan, "FELISA System Reference Manual", 1994.
- [10] Y. Kallinderis and P. Vijayan, "An Adaptive Refinement Coarsening Scheme for 3-D Unstructured Meshes", AIAA Journal, Vol 31, No.8, pp 1440-1447, Aug. 1993.
- [11] Y. Kallinderis, A. Khawaja, and H. McMorris "Hybrid Prismatic / Tetrahedral Grid Generation for Complex Geometries", AIAA Paper 95-0211, Reno, NV, January 1995.

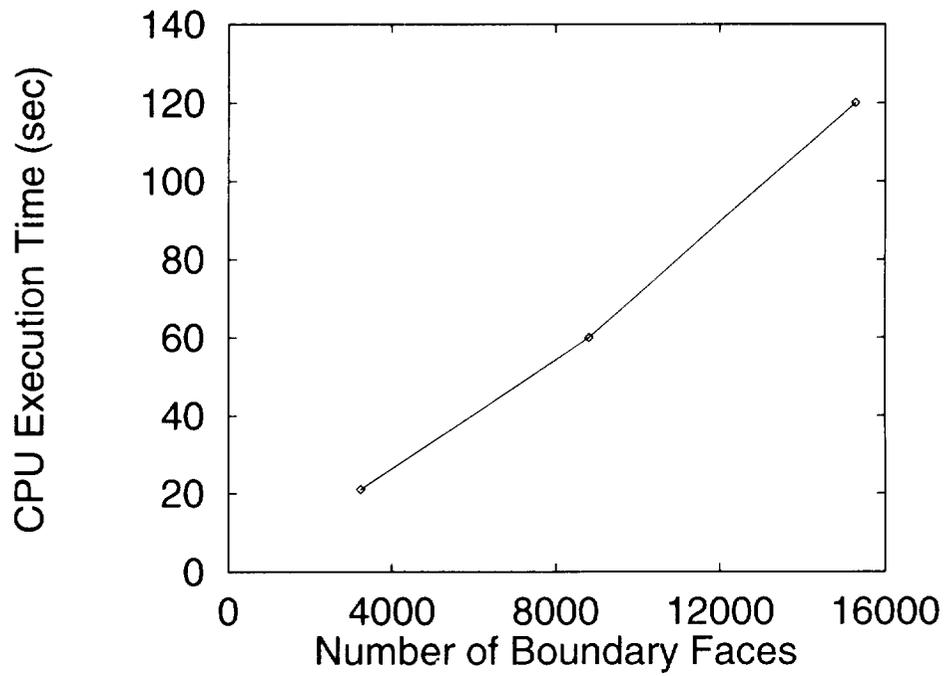


Figure 1: Scaling of prisms generation time with number of boundary faces. Case of the ONERA M6 wing with 24 layers of prisms.

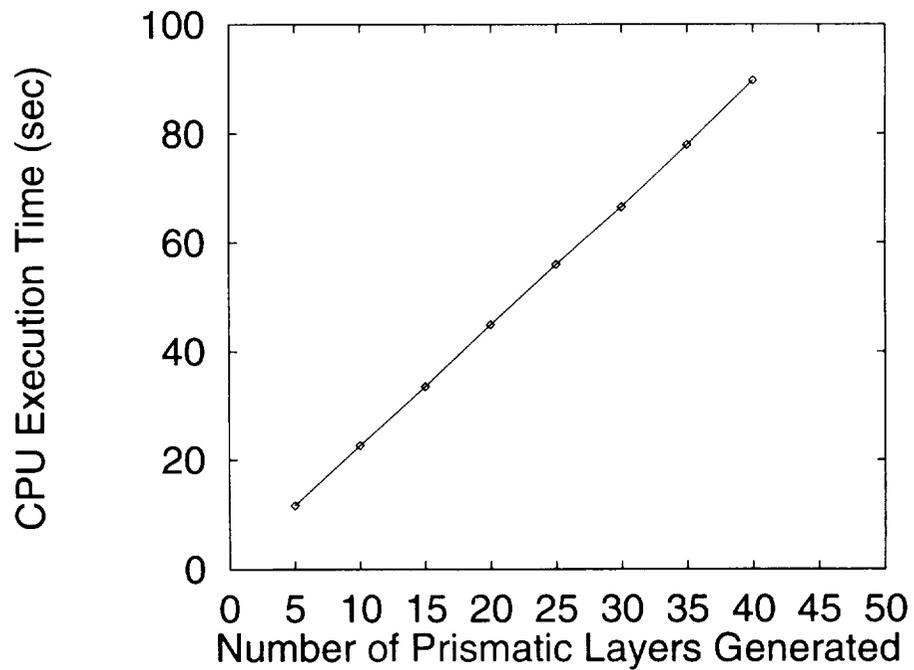


Figure 2: Linear increase in prisms generation time with number of prismatic layers. Case of the HSCT aircraft with 4412 boundary faces.

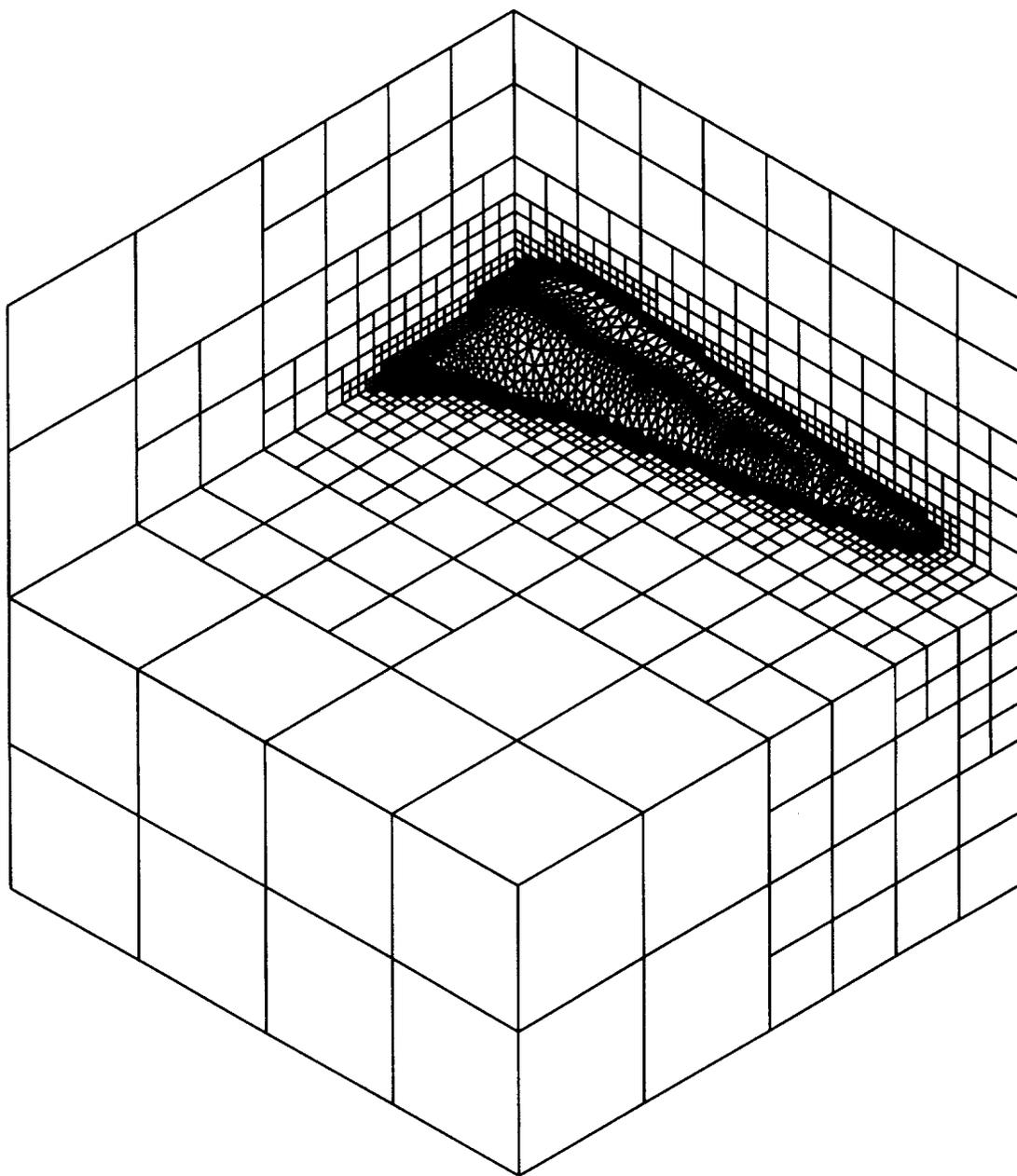


Figure 3: Outward refinement results in gradual growth of the octree.  
3-D section of domain showing the outer prismatic surface and the octants.

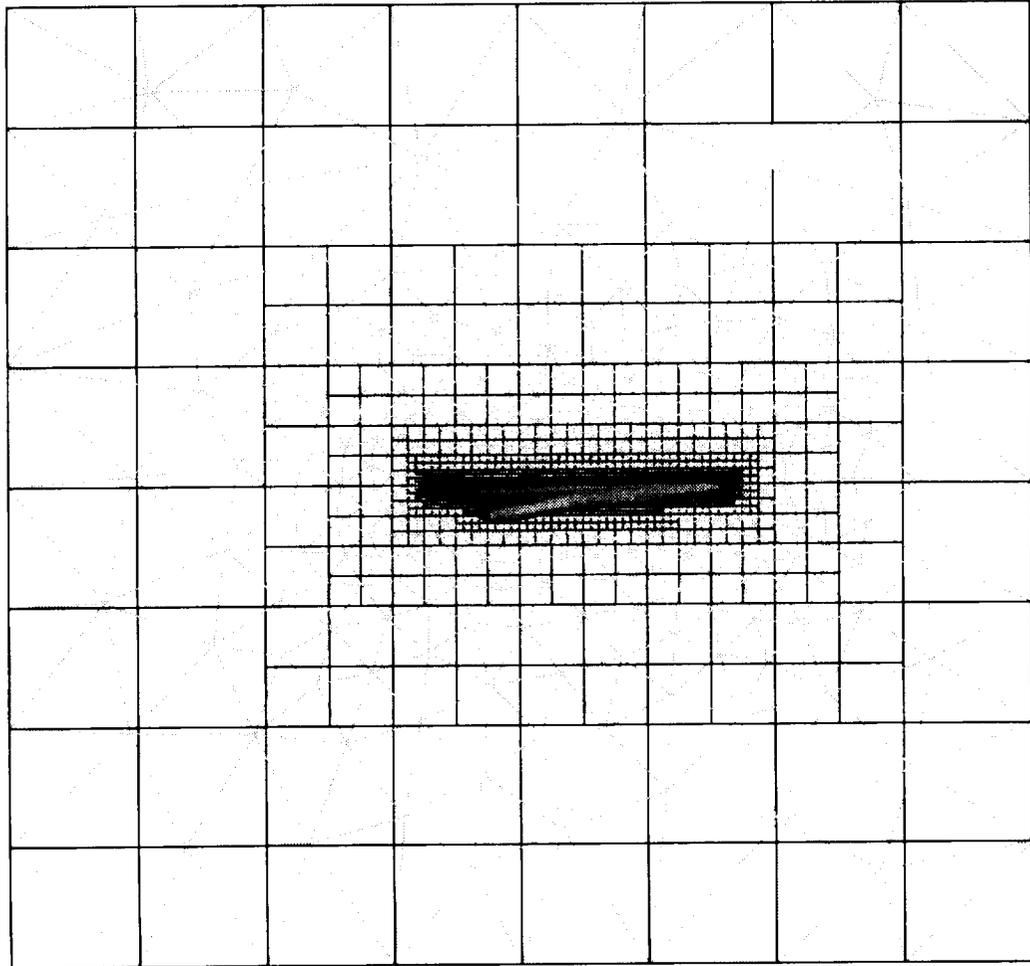


Figure 4: Effect of the octree on growth of the tetrahedra. View of the octants (quadrilateral faces), as well as of the tetrahedra (triangular faces) on the symmetry plane. Growth of the tetrahedra away from the outermost prisms surface follows growth of the octree quite faithfully.

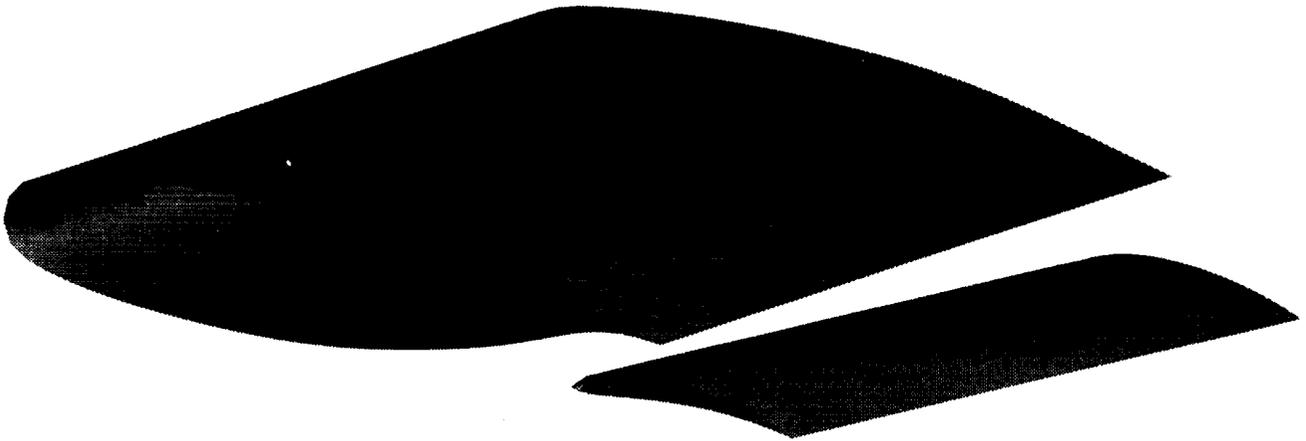


Figure 5: Geometry of two-element wing with a variable size gap between the main wing and the flap.

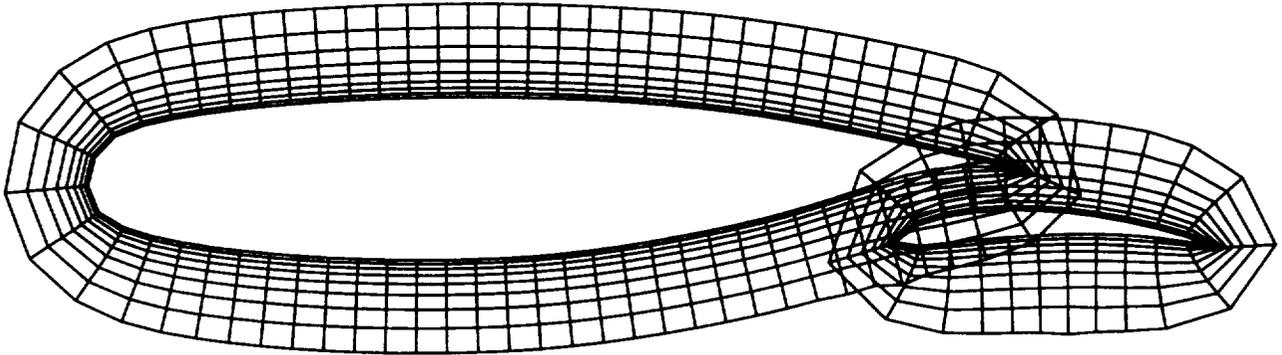


Figure 6: Prismatic grids grow around each body independently of one another (view on the symmetry plane).

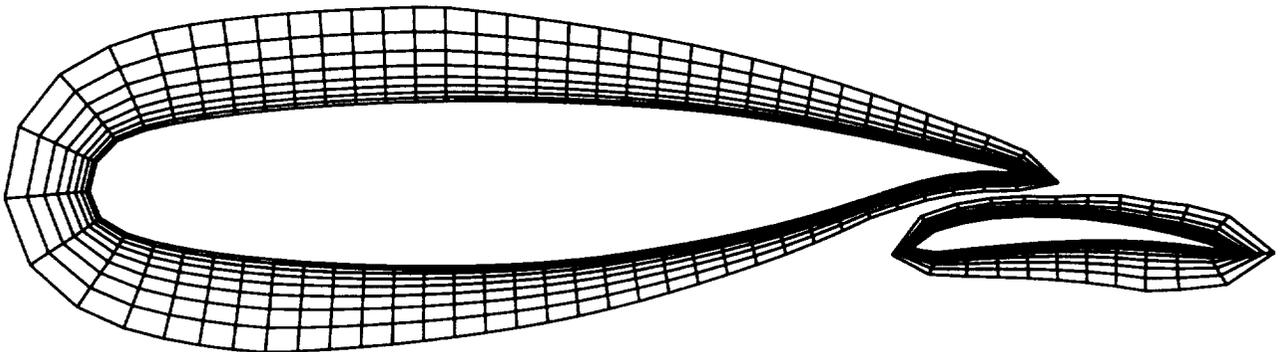


Figure 7: Mutual receding of the two prismatic grids removes prior overlapping (view on the symmetry plane).

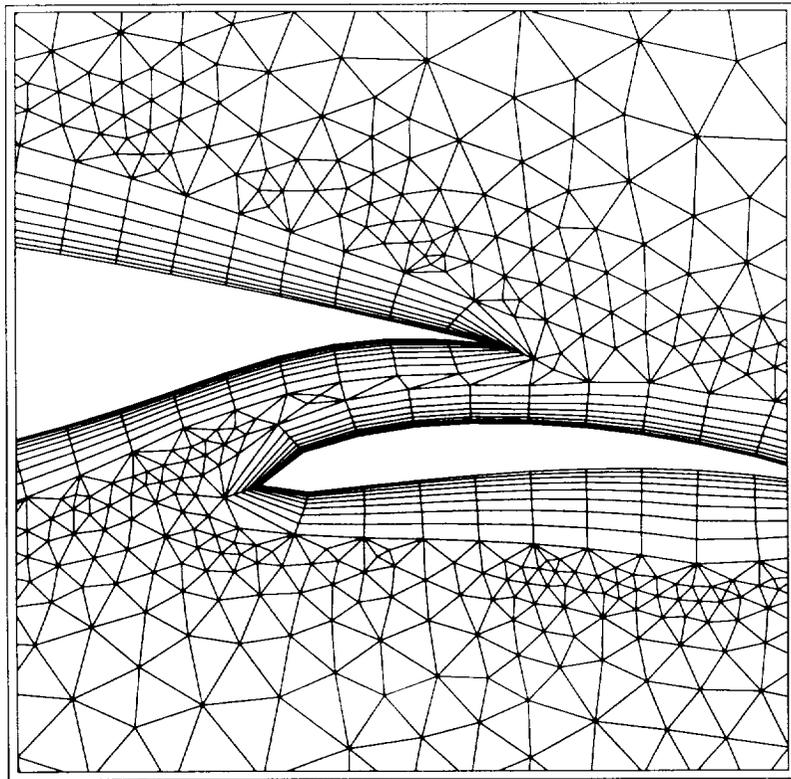
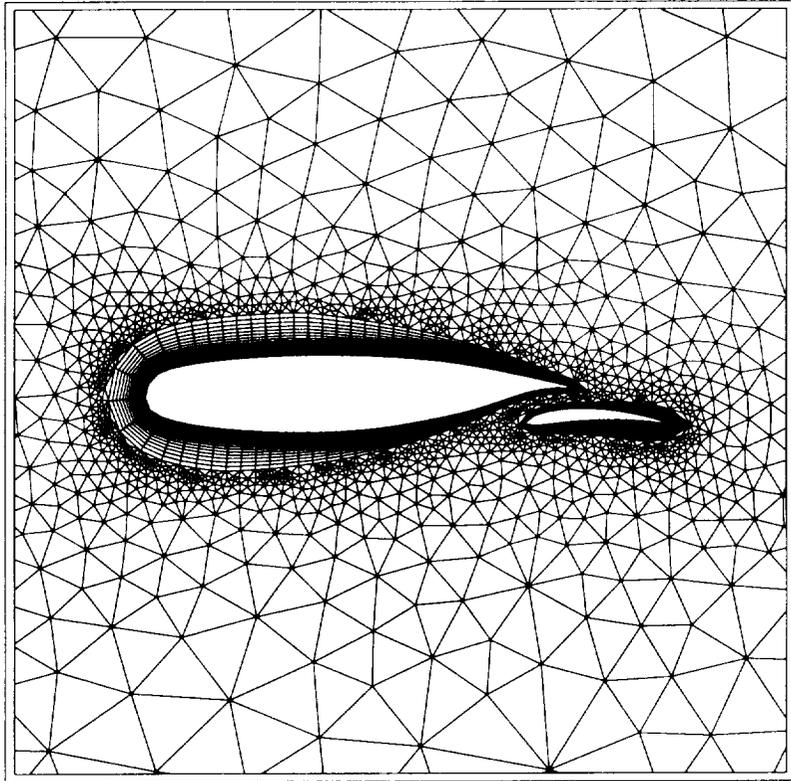


Figure 8: Tetrahedral grid fills the areas in between the two prismatic meshes (view on the symmetry plane).  
Lower portion: Enlarged view of the gap region between the two bodies.

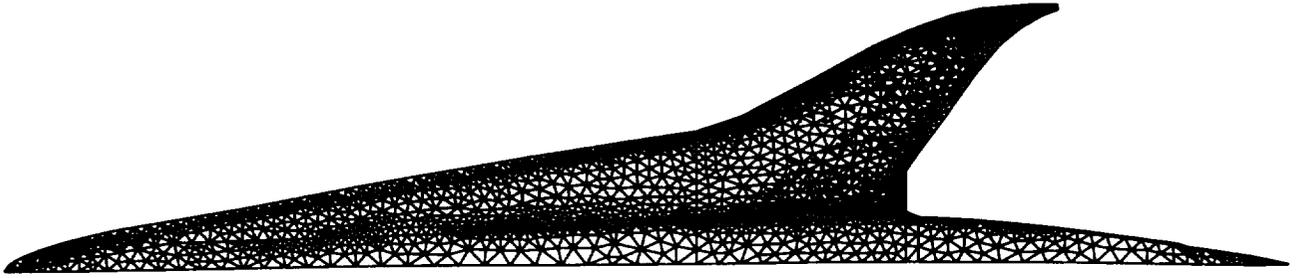


Figure 9: Triangulation of the HSCT surface (4412 triangles, 2275 nodes). A symmetry plane is considered.

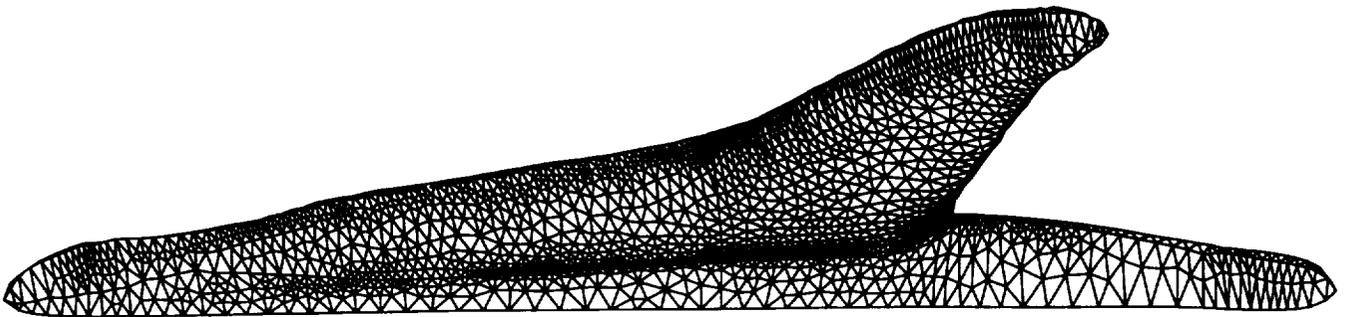


Figure 10: View of the outer surface of the prismatic mesh of the HSCT.

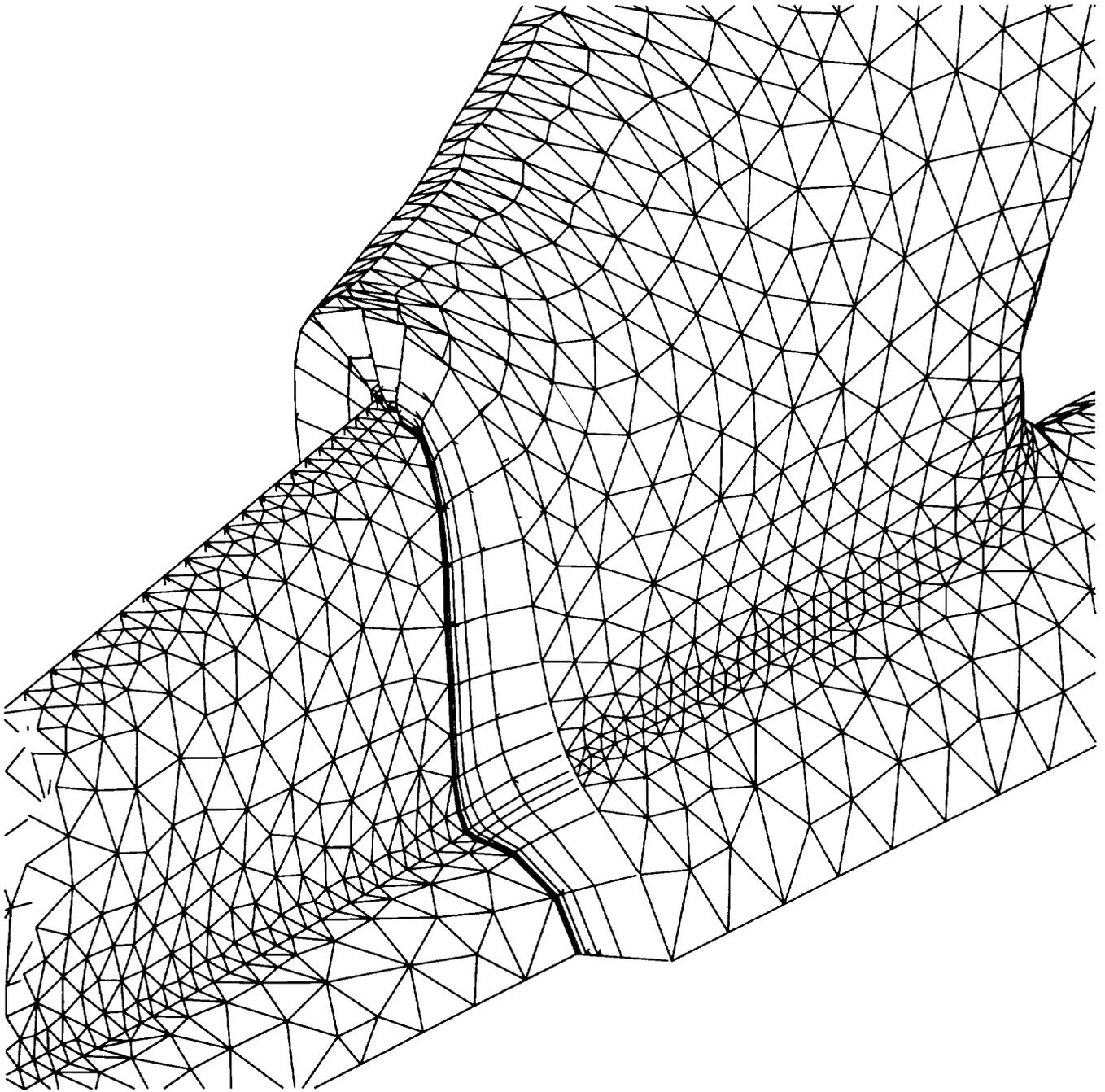


Figure 11: View of the initial and final prisms surfaces around the HST as well as of the quadrilateral faces of the prisms (every fourth layer is plotted).

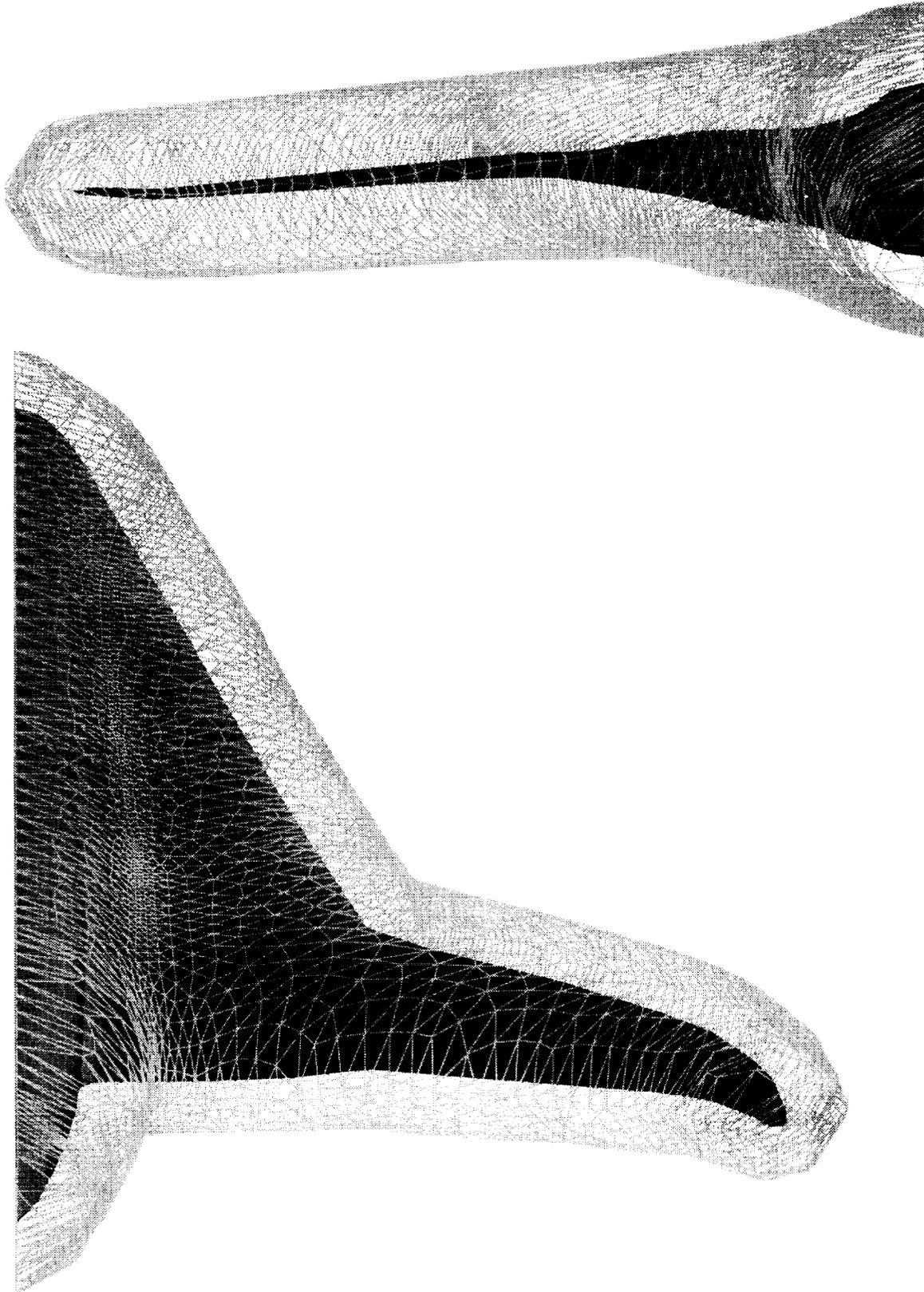


Figure 12: Wire-frame views of the grown prismatic mesh around the wing of the HSCT (shaded area denotes the surface of the aircraft).

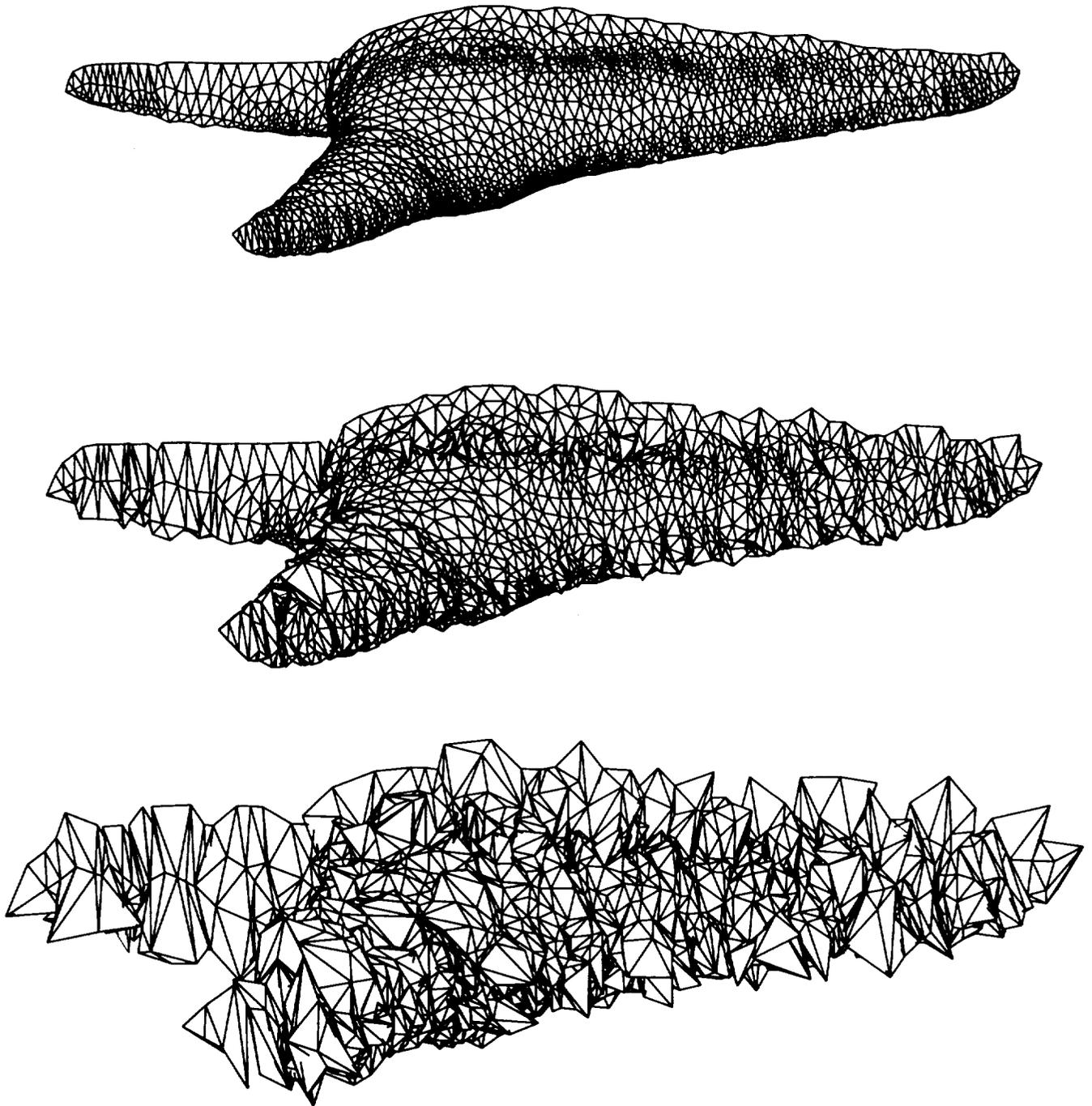


Figure 13: View of different stages in growth of the layer of tetrahedra away from the prisms outer surface for the HSCT.

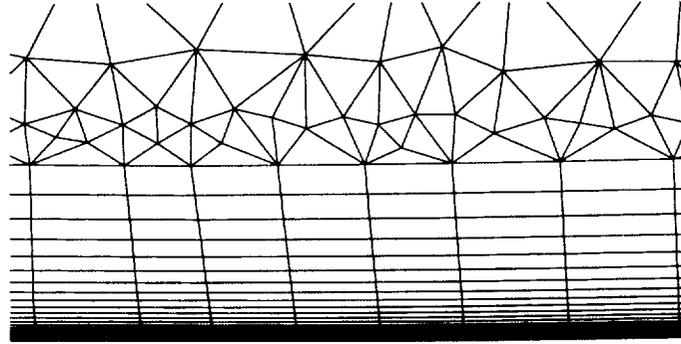


Figure 14: Smooth transition of the hybrid grid from the prisms to the tetrahedra. Partial view of the mesh on the symmetry plane.

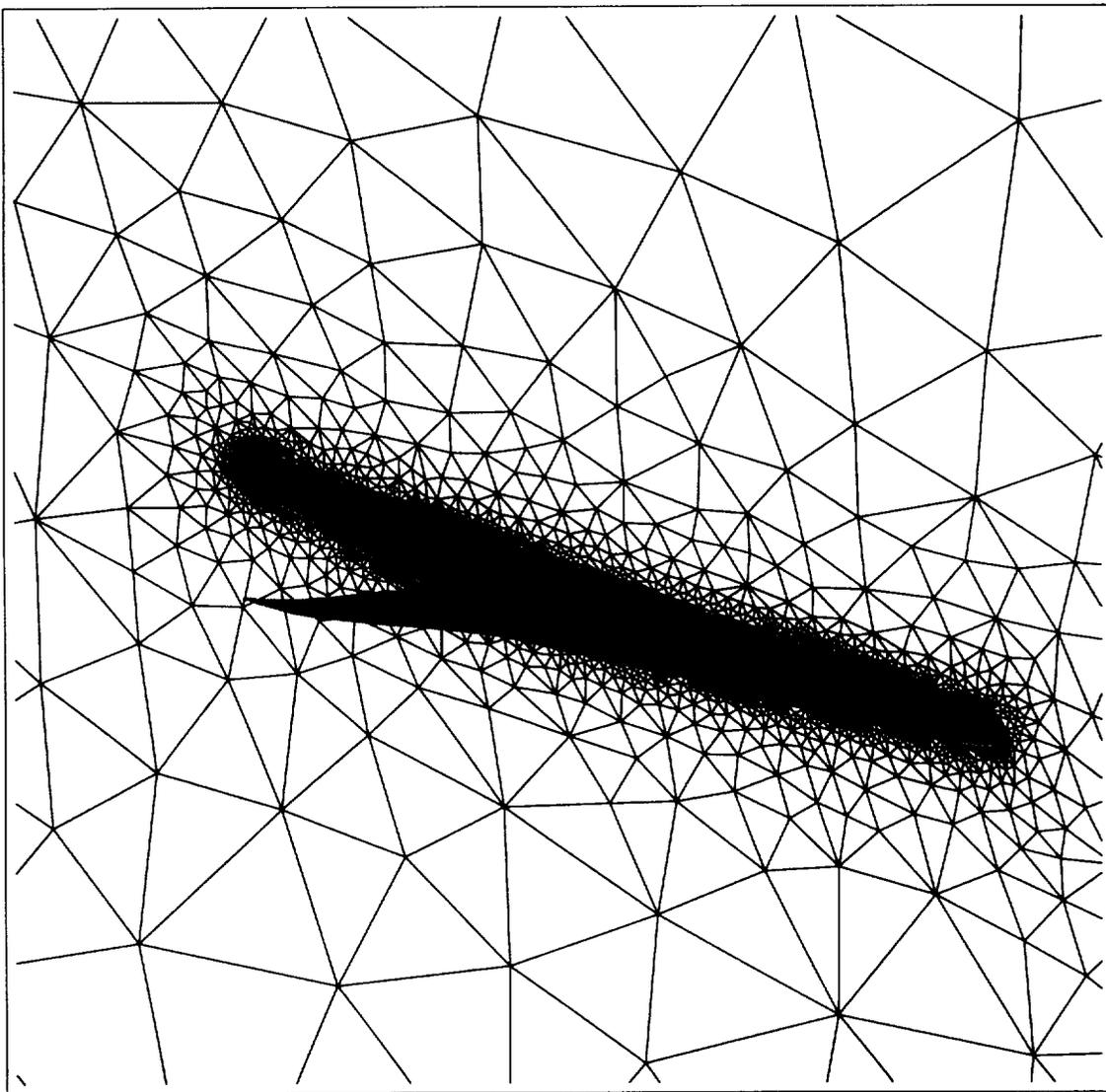


Figure 15: View of the hybrid grid on the symmetry plane of the HSCT. The quadrilateral faces correspond to the prisms, while the triangles belong to the tetrahedra.

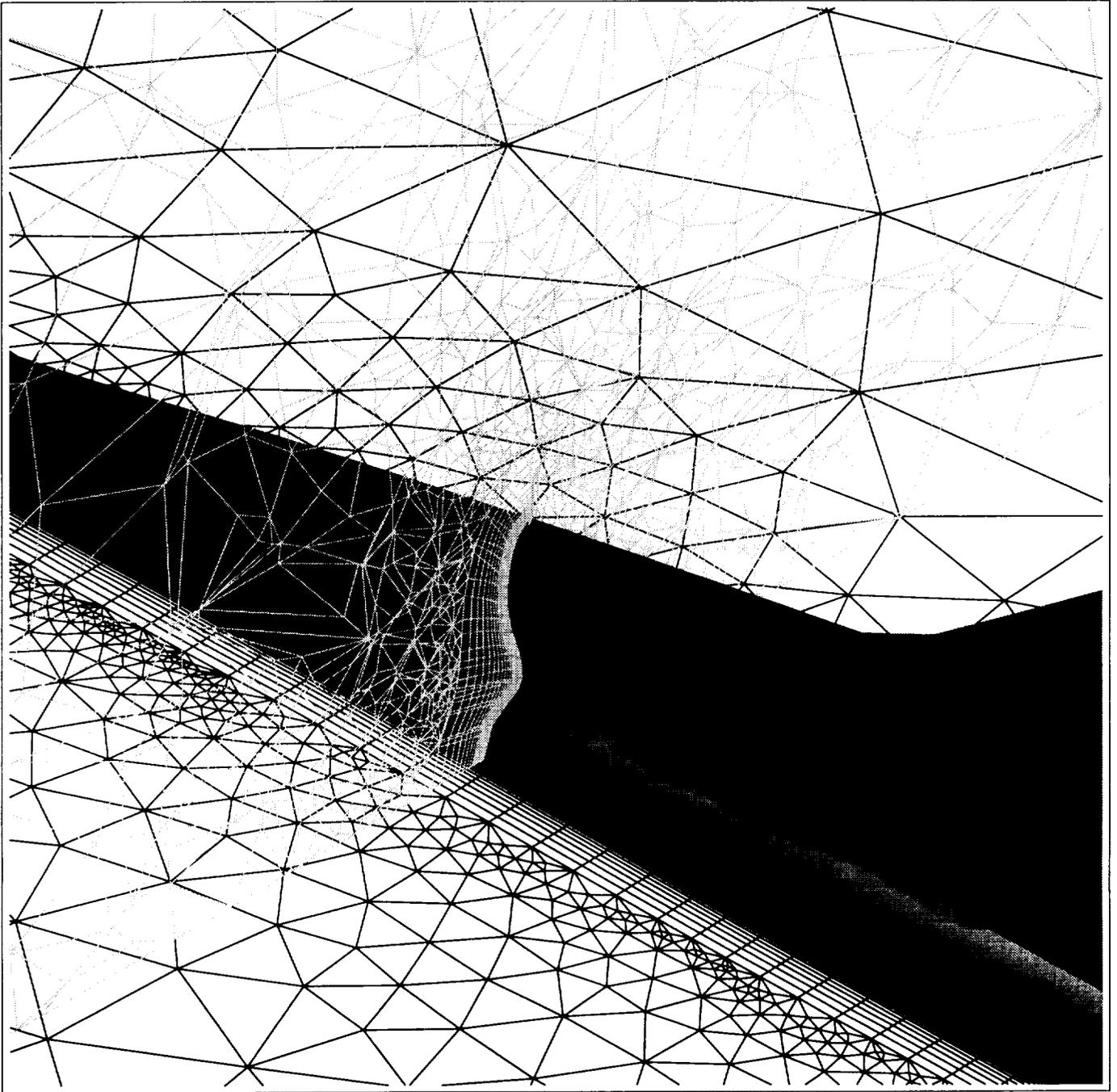


Figure 16: View of the hybrid mesh around the HSCT on two different planes that are perpendicular to each other. The first plane is the symmetry (dark lines), while the second is intersecting the fuselage at a location upstream of the wing (light lines).

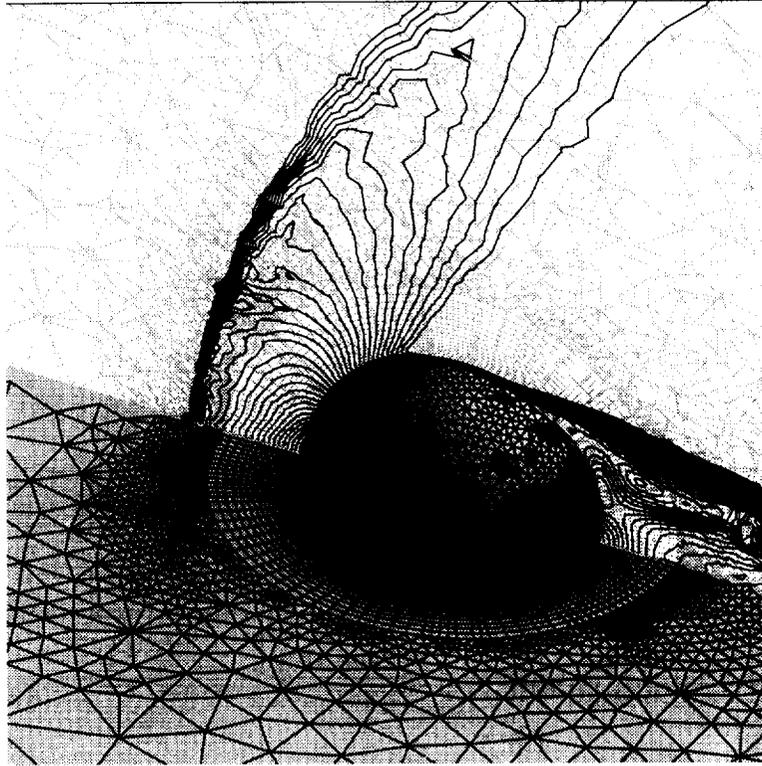


Figure 17: An isometric view of the tessellation on the wall surface, symmetry plane and an interior equatorial plane and Mach number contour lines on the equatorial plane. Hybrid grid embedded isotropically in the tetrahedral region and directionally in the prismatic region. ( $M_{min} = 0.$ ,  $M_{max} = 2.$ ,  $\Delta M = 0.05$ ).

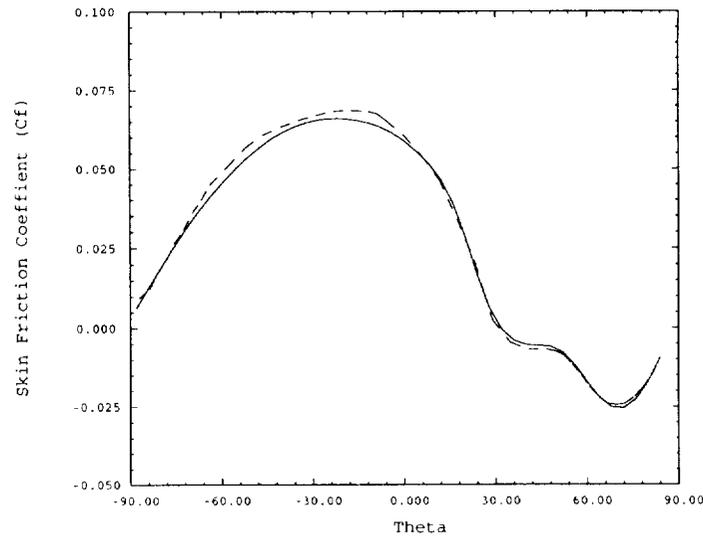


Figure 18: Comparison of Skin-friction coefficients at the wall, on the equatorial plane normal to the symmetry plane.

— Globally refined hybrid grid solution, - - - Locally embedded hybrid grid solution.